

# SEMI-AUTOMATIC SEGMENTATION OF ON-LINE HANDWRITING

**Michal Gavenčiak**

Bachelor Degree Programme (3), FEEC BUT

E-mail: [xgaven07@stud.feec.vutbr.cz](mailto:xgaven07@stud.feec.vutbr.cz)

Supervised by: Vojtěch Zvončák

E-mail: [vojtech.zvoncak@phd.feec.vutbr.cz](mailto:vojtech.zvoncak@phd.feec.vutbr.cz)

**Abstract:** This paper deals with the automation of digital trace data segmentation. The data are obtained from a digitizing tablet and are then subjected to handwriting analysis, providing quantified information about a person's handwriting, which might help in the diagnosis of handwriting difficulties. In order to successfully analyze the data, they must be segmented by individual handwriting exercise. Implementation of a python-based program with a GUI is described along with its basic functionality.

**Keywords:** Handwriting difficulties, graphical user interface, python, online handwriting

## 1 ÚVOD

Pro úspěšné absolvování školní docházky, je pro každé dítě nutné dosáhnouti dostatečné úrovně psaného projevu. V tomto ohledu se dítě považuje za zběhlé, je-li jeho psaný projev čitelný a samotné psaní dítě zvládá s minimálním úsilím a za přiměřenou dobu[5][7]. Porucha psaného projevu je velice často spojována s dětskou vývojovou dysgrafií, kterou trpí 10-30% dětí školního věku[6].

V současnosti je diagnostika této poruchy založena na vizuálním posouzení grafického projevu založeném na tvaru písmen, odsazování a počtu chyb. Tato metoda, závislá na psychickém stavu hodnotitele, jeho zraku a zkušenostech, navíc nenabízí možnost posouzení atributů psaného projevu, jako jsou například: tlak pera na papír, sklon, rychlost a zrychlení pera při dotyku s podložkou (on-surface pohyb), či pohyby perem nad papírem (in-air pohyb).

V rámci výzkumů [1][5] zabývajících se využitím moderních, digitálních metod pro zdokonalení diagnostiky této poruchy byl vyvinut postup, umožňující zachytit psaný projev pomocí digitalizačního tabletu. Při akvizici dat děti vypracovávají psaná cvičení, která jsou zadána na papíru A4, který je připevněn na tablet. Pro urychlení doby vyšetření je umístěno několik cvičení na jeden papír. Segmentace dat je prozatím prováděna manuálně s pomocí softwaru, a nabízí se tak možnost dalšího vylepšení. V rámci této práce je popsán vývoj programu, poskytující poloautomatickou segmentaci dat s možností nastavení citlivosti algoritmu a možností manuální korekce prostřednictvím uživatelského rozhraní.

## 2 SEGMENTACE DAT

Data určená ke zpracování obsahují celkem 7 číselných vektorů, přičemž každý řádek nese informaci o poloze hrotu pera (pozice  $x$ ,  $y$ ,  $z$ ) vůči digitalizujícímu tabletu, jeho náklonu, azimutu a tlaku vyvíjenému na tablet. Řádky jsou seřazeny podle časové osy v pořadí, v jakém vznikaly. Celkově tyto informace popisují jedno nebo více psacích cvičení, přičemž ve výstupních souborech z tabletu neexistuje informace o jejich počtu, poloze ani velikosti. Jelikož ke každému vzorku produkovanému tabletem získáváme i časovou značku, nazýváme toto písmo jako online.

Současné provádění ruční segmentace je náročná a zdoluhavá. Obsah souboru je nejprve nutné vykreslit pomocí specializovaného programu umožňujícího vykreslování bodů, či grafů, například MatLAB. Poté je však nutné z grafu odhadnout a přesněji určit místo v souboru, kde provést „střih“. Tento postup je, kromě velké časové náročnosti, také velmi nepřesný. Není výjimkou, že cvičení nejsou tvořena v předpokládaném pořadí. Navíc měli někteří pacienti tendenci dokreslovat některá cvičení zpětně. Při segmentaci tedy není možné spoléhat na hraniční body na časové ose, oddělující jedno cvičení od druhého.

## 2.1 PROGRAM DYSGRAPHY

Program DysGraphy byl vyvinut, aby usnadnil segmentaci dat. Především tedy, aby minimalizoval čas potřebný ke zpracování jednoho souboru a maximalizoval přesnost, se kterou je operace prováděna. Použitým programovacím jazykem je Python 3.7.0. Program dále využívá volně dostupné knihovny:

- NumPy[2] - práce s vektory, maticemi a nástroje pro správu těchto a jiných dat.
- Matplotlib[3] - vykreslování různých typů 2D a 3D grafů a obrazců s možností jednoduchých ovládacích prvků a podporou vnoření do grafického uživatelského rozhraní.
- PyQt5 - tvorba grafického uživatelského rozhraní.

Tyto prostředky byly zvoleny především na základě vývojové složitosti. Pro vývoj programu byla také posuzována implementace v programovacím jazyce C++. Program by jistě nabízel větší výkon a rychlejší zpracování, nicméně tvorba kódu v C++ je daleko složitější a zdoluhavější[4], což ubírá z času, který může být věnován vývoji dalších funkcí. Cílem tohoto programu navíc není dosáhnout co nejvyššího výkonu.

## 2.2 AUTOMATIZOVANÁ SEGMENTACE

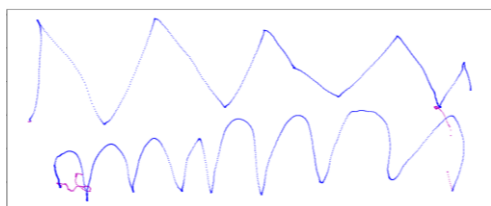
Data pro segmentaci jsou načtena kompletně ze souboru do paměti počítače. Samotné zpracování informací se nyní dělí na několik kroků.

Prvním krokem je rozdělení dat na jednotlivé tahy na papíře (stroke) a pohyby nad papírem (off-stroke). K docílení tohoto rozdělení program projde všechna data dle času a podle informace o přiložení pera na papír (1, 0 v každém řádku dat) oddělí úseky na jednotlivé instance tříd stroke a off-stroke (viz obrázek 2).

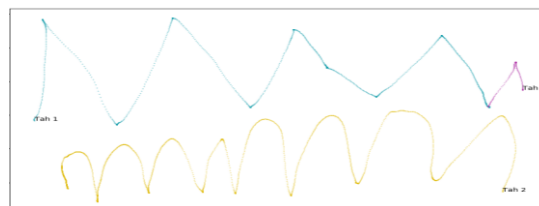
Druhým krokem je rozpoznání souvisejících tahů, které dohromady tvoří jedno cvičení. K tomuto účelu je využita analýza pozice a velikosti jednotlivých tahů. Nejprve je porovnávána jejich vzájemná poloha a ty, které jsou vyhodnoceny jako blízké, jsou označeny jako potenciálně související. Pro každý pár takto podezřelých je vyhodnocena poloha, kde je kritické přiblížení tahů nejpravděpodobnější. Kritickým přiblížením je myšleno protnutí dvou tahů, nebo takové přiblížení, kdy je jasné, že tahy spolu souvisí, např. nedotažení nožičky při psaní písmene „T“. V této lokalitě jsou nyní porovnávány pozice jednotlivých bodů tahů. Vyhodnocení souvislosti probíhá dále také detekcí vepsaných a opsaných tahů. Pokud by předmětem některého cvičení bylo nakreslit například, program vyhodnotí vnitřní i vnější kružnici jako spolu související, i přes to, že nejsou v kritické blízkosti. Před započítáním běhu algoritmu je možné nastavit jeho citlivost. Nastavení citlivosti pomocí číselné hodnoty, udávané v pixelech, ovlivňuje vzdálenosti, v rámci tohoto kroku, které jsou vyhodnoceny jako kritické, či související (viz obrázek 3).

Třetím krokem je rozřazení nalezených párů do jednotlivých cvičení, které v rámci programu zastupují instance třídy nazvané *Picto* (zkráceně z ang. pictogram). Páry obsahující jeden shodný tah jsou sloučeny do jedné instance této třídy, přičemž je důležité, aby nevznikaly duplicitní prvky, které by mohly zkreslit výsledky analýzy po segmentaci dat. Tahy, které nejsou spárovány, jsou rozřazeny samostatně, neboť cvičení mohou být jedno-tahová (viz obrázek 4).

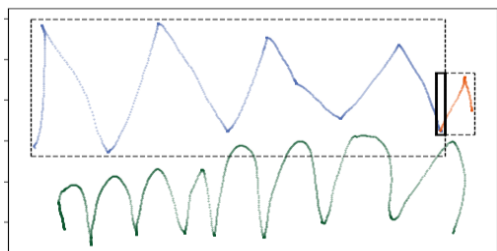
(Pozn. autora: z obrázků 2, 3 a 4 bylo pro přehlednost odstraněno vykreslení pohybů nad papírem)



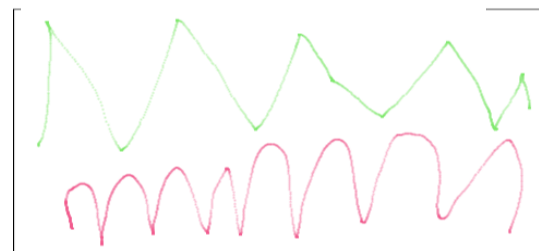
**Obrázek 1:** Vykreslení výstupních dat z tabletu: modrá (on-surface): fialová (in-)



**Obrázek 2:** Vykreslení dat po segmentaci na jednotlivé tahy



**Obrázek 3:** Vzájemné porovnání dvou tahů vyznačených čárkovaně, oblast prohledávaná pro kritickou blízkost tučně



**Obrázek 4:** Vykresleny dvě nalezená, oddělená cvičení

### 2.3 RYCHLOST ALGORITMU

V průběhu vývoje došlo k několikanásobnému zrychlení algoritmu. Použijme jako referenční hodnotu dobu, kterou zpracovávala první verze algoritmu největší ze vzorkových souborů (31830 řádků, 1,127 KB, 10 cvičení, 49 tahů), tedy přibližně 33,6 s. Implementací postupného vyřazování podezřelých párů a hledání protnutí, či kritického přiblížení tahů pouze ve vymezené lokalitě, došlo ke zkrácení tohoto času na 11,2 s. Další optimalizací zacházení s daty v průběhu chodu algoritmu došlo k dalšímu zrychlení na 5,0 s. Většina vzorkových souborů je však menší a obsahuje přibližně 5000 – 10000 řádků. U souborů této velikosti se doba zpracování pohybuje kolem 1 s.

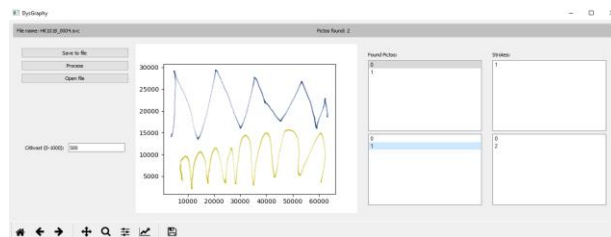
### 2.4 SPOLEHLIVOST ALGORITMU

Přesnost algoritmu je závislá na nastavení jeho citlivosti. Ta je defaultně nastavena na hodnotu 500 px. V rámci testování, však ještě nebyla nalezena hodnota, která by se dala považovat za ideální, nebo univerzální. Obecně lze vzorové soubory rozdělit do tří kategorií: A – soubory, které jsou správně vyhodnoceny s tímto nastavením. B – soubory, které vyžadují nastavení citlivosti výrazně nižší, neboť cvičení v nich jsou na sebe příliš natěsnána. C – soubory vyžadující hodnotu vyšší, neboť mezery způsobené nedotahováním jsou příliš velké.

Při špatném vyhodnocení je možné změnit hodnotu citlivosti a provést zpracování znovu, případně provést manuální korekci v rámci grafického rozhraní.

### 2.5 OVLÁDÁNÍ PROGRAMU

Funkce je možné ovládat prostřednictvím jednoduchého uživatelského rozhraní. To nabízí možnosti volby souboru pro načtení dat, složky pro uložení zpracovaných dat, nastavení citlivosti algoritmu, spuštění algoritmu a případně následnou korekci špatně vyhodnocených cvičení. Korekce se provádí zvolením dvou odpovídajících cvičení ze seznamu a přetažením špatně zařazeného tahu ke správnému cvičení pomocí funkce Drag and Drop. GUI dále obsahuje pole pro vykreslení surových dat a segmentovaných cvičení včetně kroků korekce (viz obrázek 5).



**Obrázek 5:** Grafické rozhraní programu DysGraphy

## 2.6 DALŠÍ VÝVOJ

V tuto chvíli je program funkční a je schopen splnit účel, pro který byl navržen. Beze sporu však nabízí spoustu prostoru pro další vývoj. Z hlediska chodu algoritmu je možné provést další optimalizaci, kterou dojde ke zrychlení chodu programu. Nejdůležitějším bodem dalšího vývoje však bude rozšíření grafického rozhraní, které i když poskytuje všechny potřebné nástroje, není moc uživatelsky přívětivé a celkově je spíše minimalistické.

Rád bych také zvážil možnost implementovat algoritmus použitý k následné analýze písma, přímo do tohoto programu, aby bylo výsledné softwarové řešení co nejkompaktnější.

## 3 ZÁVĚR

Tento článek popisuje implementaci, chod a funkce programu DysGraphy, který byl vytvořen pro usnadnění přípravy dat, obsahujících on-line písmo, pro další analýzu. Implementovaný algoritmus zpracuje běžně veliký soubor za přibližnou dobu 1 s a je možné měnit jeho citlivost. Program disponuje grafickým rozhraním s ovládacími prvky, které nabízí možnost náhledu a korekce vyhodnocení programu.

## REFERENCE

- [1] MEKYSKA, JIRI, MARCOS FAUNDEZ-ZANUY, ZDENEK MZOUREK, ZOLTAN GALAZ, ZDENEK SMEKAL and SARA ROSENBLUM. Identification and Rating of Developmental Dysgraphia by Handwriting Analysis. *IEEE Transactions on Human-Machine Systems* [online]. 2017, vol. 47, no. 2, pp. 235-248 [accessed. 13 . December 2018]. Retrieved z: doi:10.1109/thms.2016.2586605
- [2] OLIPHANT, TRAVIS E. *Guide to NumPy*. B.m.: USA: Trelgol Publishing, 2018
- [3] HUNTER, JOHN D. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering* [online]. 2007, vol. 9, no. 3, pp. 90-95 [accessed. 13 . December 2018]. Retrieved z: doi:10.1109/mcse.2007.55
- [4] PRECHELT, LUTZ. An empirical comparison of C, C++, Java, Perl, Python, REXX, and Tcl for a search/string-processing program. 2000.
- [5] ZVONČÁK, V.; MEKYSKA, J.; ŠAFÁROVÁ, K.; GALÁŽ, Z.; MUCHA, J.; KISKA, T.; SMÉKAL, Z.; LOSENICKÁ, B.; ČECHOVÁ, B.; FRANCOVÁ, P.; FAUNDEZ-ZANUY, M.; ROSENBLUM, S. Effect of Stroke-level Intra-writer Normalization on Computerized Assessment of Developmental Dysgraphia. In 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). Moskva, Rusko: 2018. p. 233-237. ISBN: 978-1-5386-9360-5
- [6] KUSHKI, AZADEH, HEIDI SCHWELLNUS, FAIZAH ILYAS and TOM CHAU. Changes in kinetics and kinematics of handwriting during a prolonged writing task in children with and without dysgraphia. *Research in Developmental Disabilities* [online]. 2011, vol. 32, no. 3, pp. 1058-1064. Retrieved z: doi:10.1016/j.ridd.2011.01.026
- [7] VAN GEMMERT, AREND W. A., HANS-LEO TEULINGS. Advances in graphonomics: Studies on fine motor control, its development and disorders. *Human Movement Science* [online]. 2006, **25**(4-5), 447-453 [cit. 2019-03-15]. DOI: 10.1016/j.humov.2006.07.002. ISSN 01679457. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S0167945706000595>